

**VŠB – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra měřicí a řídicí techniky**

Výuková aplikace pro zpracování signálu a soustav  
Learning application for signal and system analysis

# Zadání bakalářské práce

Student: **Roman Slabý**

Studijní program: B2645 Elektrotechnika, sdělovací a výpočetní technika

Studijní obor: 2612R041 Řídicí a informační systémy

Téma: **Výuková aplikace pro zpracování signálů a soustav**  
**Learning application for signal and system analysis**

Zásady pro vypracování:

1. Rozbor problematiky uživatelského rozhraní v Matlabu
2. Návrh uživatelského rozhraní v Matlabu
3. Řešení uživatelského rozhraní pro výuku zpracování signálů a soustav
4. Srovnání naměřených výsledků s teoretickými předpoklady
5. Zhodnocení dosažených výsledků

Seznam doporučené odborné literatury:

1. SEDLÁČEK, Miloš, ŠMÍD, Radislav. Matlab v Měření. Praha : ČVUT, 2005. 204 s.
2. NEVŘIVA, Pavel. Simulace řídicích systémů na číslicovém počítači. 1975. vyd. Praha 1 : SNTL, 1975. 136 s.
3. NEVŘIVA, Pavel. Analýza signálů a soustav. [s.l.] : Ben, 2000. 672 s.
4. Manuály a příručky on-line od firmy Humusoft

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Zdeněk Macháček, Ph.D.**

Datum zadání: 30.11.2008

Datum odevzdání: 07.05.2009

---

prof. Ing. Vilém Srovnal, CSc.  
*vedoucí katedry*

---

prof. Ing. Ivo Vondrák, CSc.  
*děkan fakulty*

## **Prohlášení studenta**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě.....

Podpis studenta

Prohlašuji, že

- jsem byl seznámen s tím, že na moji bakalářskou/diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména §35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a §60 – školní dílo.
- beru na vědomí, že Vysoká škola báňská – technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně ke své vnitřní potřebě bakalářskou/diplomovou práci užít (§35 ods. 3).
- souhlasím s tím, že jeden výtisk bakalářské/diplomové práce bude uložen v Ústřední knihovně VŠB-TUO k prezenčnímu nahlédnutí a údaje o bakalářské /diplomové práci budou zveřejněny v informačním systému VŠB-TUO.
- beru na vědomí, že odevzdáním své práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, bez ohledu na výsledek její obhajoby.

V Ostravě.....

Podpis studenta

## **Poděkování**

Rád bych poděkoval vedoucímu bakalářské práce Ing. Zdeňku Macháčkovi, Ph.D., za jeho odborné rady a cenné připomínky , které mi byly v průběhu této práce poskytnuty. Dále děkuji Ing. Janu Krňávkovi a Ing. Martinovi Piešovi za poskytnutí praktických rad.

**Abstrakt:**

Tato práce se zabývá tvorbou výukové aplikace pro zpracování signálů v prostředí Matlab. Aplikace umožňuje základní zpracování signálu. V této souvislosti se rozumí generování signálu na výstup zvukové a měřicí karty, zobrazení signálu ze vstupu zvukové a měřicí karty. Zpracování signálu je analýza, výklad a manipulace získaných signálů. Znalost vlastností signálů je nutná pro návrh řídicích systémů. V praxi se rozlišuje analogové nebo digitální zpracování. Digitální zpracování je výhodné zejména díky jeho odolnosti vůči rušení.

**Klíčová slova:**

Analýza signálu, Matlab, MF 624, zvuková karta, vzorkovací frekvence

**Abstract:**

This work deals with the creation of educational applications for signal processing in Matlab. The application allows the basic signal processing. In this context, it means generating a signal to the output and measuring the sound card, display of the input signal and measuring the sound card. Signal Processing is the analysis, interpretation and manipulation of acquired signals. Knowledge of the characteristics of signals is required for the design of control systems. In practice, a distinction is made between analog or digital processing. Digital processing is particularly beneficial due to its resistance to interference.

**Key word:**

Analysis of signal, Matlab, MF 624, sound card, sampling rate

## Seznam použitých symbolů a zkratek:

$A$	zesílení
$A_1$	průměrné zesílení
$B$	šířka pásma vzorkovaného signálu, [Hz];
$E$	energie signálu, [J];
$f$	přepočítaná frekvence signálu, [Hz];
$f_g$	frekvence generovaného signálu, [Hz];
$P$	střední výkon signálu, [W];
$T$	perioda, [s];
$t$	proměnná času, [s];
$U_a$	zadaná hodnota napětí v aplikaci, [V];
$U_g$	generovaná hodnota napětí, [V];
$U_z$	změřená hodnota napětí, [V];
$U_o$	odchylka napětí, [V];
$U_{o\%}$	odchylka napětí, [%];
$w(t)$	signál v časovém intervalu
$w_m$	koeficienty Fourierovy řady
$W(\omega)$	spektrální funkce
$\omega_s$	vzorkovací kmitočet signálu
$\varphi$	počáteční fáze

GUI	Grafické uživatelské rozhraní
PCI	Peripheral Component Interconnect, počítačová sběrnice pro připojení periférií k základní desce
TTL	tranzistorově-tranzistorová logika
RTT	Real Time Toolbox

## Obsah:

<b>1. Úvod.....</b>	<b>1</b>
<b>2. Programové prostředí Matlab .....</b>	<b>2</b>
2.1. Matlab .....	2
2.2. Simulink .....	3
2.3. Grafické uživatelské rozhraní .....	3
2.4. Grafické objekty.....	4
<b>3. Teorie analýzy signálů.....</b>	<b>5</b>
3.1. Výkonový signál .....	5
3.3. Shannon- Kotělníkův teorém .....	5
3.4. Fourierova transformace .....	5
3.5. Fourierova řada .....	6
<b>4. Měřicí karta MF 624 .....</b>	<b>7</b>
4.1. Technické parametry.....	7
4.2. Real Time Toolbox .....	8
<b>5. Zvuková karta .....</b>	<b>9</b>
<b>6. Realizace výukové aplikace .....</b>	<b>10</b>
6.1. Struktura aplikace .....	10
6.2. Modelování signálů.....	15
6.2.1. Výkonový signál.....	15
6.2.1.1. Harmonický signál .....	15
6.2.1.2. Obdélníkový signál .....	16
6.2.1.3. Pilový signál.....	16
6.2.2. Energetický signál .....	18
6.2.2.1. Harmonický signál .....	18
6.2.2.2. Obdélníkový signál .....	18
6.2.2.3. Pilový signál.....	19
6.3. Generování a měření signálů zvukovou kartou.....	20
6.3.1. Generování signálu.....	20
6.3.2. Měření a zobrazení signálu.....	26
6.4. Generování a měření signálů měřicí kartou MF624 .....	28
6.4.1. Generování signálu .....	28
6.4.2. Zobrazení signálu .....	30
<b>7. Shrnutí výsledků a závěr .....</b>	<b>33</b>
 <i>Literatura:</i> .....	 34
<i>Přílohy:</i> .....	35

## 1. Úvod

Tato práce se zabývá vytvořením výukové aplikace pro zpracování signálů. Signálem je nazývána fyzikální veličina měnící se v čase. Nejběžnější signály technické praxe jsou teplotní změny, elektrické napětí, elektrický proud. Pojmem signál se rozumí nosič informací, ať už jako signál analogový nebo digitální. Informace o vlastnostech signálu se získávají analýzou signálů. Čím více je těchto informací, tím lépe lze signál popsat matematickým popisem. Tímto popisem lze signál vyjádřit v časové nebo frekvenční oblasti. Větší přesnost při dalším zpracování signálu je dosažena vyšším vzorkováním. Díky novým technologiím se zvýšila vzorkovací frekvence a tím lze signál vzorkovat vyšším počtem bodů. Při vzorkování musí být splněn Shannon- Kotělníkův teorém, aby bylo možné signál zpětně obnovit.

Aplikace by měla sloužit studentům k lepšímu pochopení dané problematiky. Měli by být schopni popsat signál v časové a frekvenční oblasti. Studenti si budou moci vybrat mezi výkonovým nebo energetickým signálem. V každé této nabídce je možnost z výběru harmonického, obdélníkového nebo pilového průběhu.

Pro vytvoření aplikace bylo použito grafické uživatelské rozhraní programového prostředí Matlab. Díky tomuto rozhraní se pracuje s grafickými prvky, které je možno přesouvat po pracovní ploše. Tím je značně ulehčena práce programátora.

První kapitola „Úvod“ seznamuje s bakalářskou prací.

Druhá kapitola popisuje programovací prostředí Matlab, způsob práce v grafickém uživatelském rozhraní včetně způsobu změny vlastností objektu uicontrol v úrovni Figure.

Třetí kapitola je zaměřena na teorii analýzy signálů. Jsou zde popsány všechny použité předpoklady k úspěšnému naprogramování aplikace.

Čtvrtá kapitola se zabývá popisem měřicí karty MF 624 a způsob komunikace s programem Matlab. Tato karta se v aplikaci používá pro generování a zobrazování signálu.

Pátá kapitola seznamuje se zvukovou kartou používanou v aplikaci pro generování a zobrazení signálu.

Šestá kapitola popisuje strukturu aplikace. Aplikace je rozdělena na hlavní soubor a funkce, řešící danou problematiku. Dále je zde znázorněna hierarchie obrazovek včetně jejich screenů.

Poslední, sedmá kapitola s názvem „Závěr“ obsahuje dosažené výsledky a další návrhy na zdokonalení výtahu.



## 2. Programové prostředí Matlab

### 2.1. Matlab

Název Matlab vznikl z anglického Matrix Laboratory. Matlab je programové prostředí a skriptovací jazyk určený zejména těm, kteří potřebují řešit početně náročné úlohy, zejména vědeckotechnické výpočty, modelování, simulace a zpracování signálů. Stal se celosvětovým standardem ve sféře vědy, výzkumu a průmyslu, ale i v oblasti vzdělávání.

Matlab má otevřenou architekturu, díky které je možné rozšíření použití programu v příslušných vědních a technických oborech. Tyto knihovny je možno rozšiřovat, modifikovat nebo jen čerpat informace z přehledně dokumentovaných algoritmů. Díky otevřené architektuře vzniklo hodně firem, které se zabývají vývojem a distribucí vlastních produktů, které buď rozšiřují výpočetní prostředí o další knihovny a nástroje nebo zajišťují propojení Matlabu s jinými specializovanými programy.

Matlab byl implementován na všech významných platformách jako je Windows, Linux, Solaris a Mac.

Systém Matlab nabízí:

- rychlé výpočetní jádro
- podpora paralelních výpočtů
- otevřený a rozšiřitelný systém
- působivá 2D a 3D grafika
- konfigurovatelné uživatelské rozhraní Matlab Desktop
- velké množství aplikačních knihoven
- programovací jazyk 4. generace
- objektové programování
- integrace s jazykem Java
- podpora vícerozměrných polí a uživatelsky definovaných datových struktur
- interaktivní nástroje pro tvorbu grafického uživatelského rozhraní
- podpora řádkových matic
- interaktivní průvodce importem dat
- zvukový vstup a výstup, animace
- komunikace s externím přístrojovým vybavením (sériová linka, GPIB, VISA)
- výpočetní jádro pro programy psané ve Fortranu a jazyce C
- distribuce nezávislých uživatelských aplikací: překlad do jazyka C, runtime modul, WWW technologie
- rozsáhlá tištěná i hypertextová on-line dokumentace

Programovací jazyk obsahuje všechny nezbytné příkazy, jako jsou podmíněné příkazy, větvící příkazy, cykly, díky kterým je možné vyvíjet i velice složité aplikace.

Předností Matlabu je také jeho integrace s jazykem Java. Skripty vytvořené v tomto jazyce je možné použít i v Matlabu, což umožňuje vytváření složitých grafických rozhraní s použitím grafických objektů Javy a také využití dostupných knihoven, které byly v jazyce Java vytvořeny.

#### Rozdělení Matlabu

- Matlab
- Rozšíření Matlabu
- Toolboxy
- Simulink
- Blocksets

[1], [2]

## 2.2. Simulink

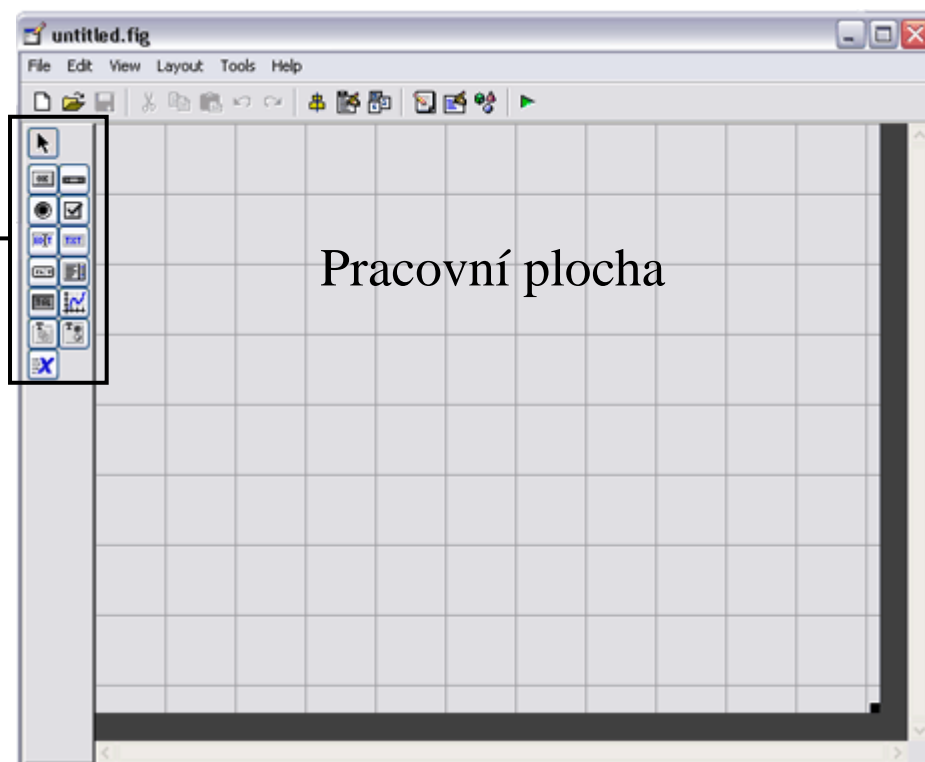
Simulink je rozšiřující součást Matlabu, určený pro simulaci a modelování dynamických systémů (systému proměnných v čase), který využívá algoritmy matlabu pro numerické řešení nelineárních diferenciálních rovnic. Simulink poskytuje ve formě blokových schémat a rovnic snadnou a rychlou tvorbu modelů dynamických soustav. Součástí Simulinku je grafický editor sloužící k vytváření modelů lineárních, nelineárních, v čase diskrétních nebo spojitých systémů pouhým přesouváním funkčních bloků. Stejně jako samotný Matlab má otevřenou architekturu, která dovoluje uživateli vytvářet vlastní funkční bloky a rozšiřovat již tak bohatou knihovnu Simulinku.

Interaktivní způsob tvorby a simulace modelů se spouští z příkazové řádky systému Matlab, příkazem simulink nebo ikonkou Simulinku v titlebaru. Po spuštění je vytvořeno okno pro tvorbu nového modelu a okno obsahující základní nabídku otvírání knihoven zdrojů signálů, základních spojitých, diskrétních a nelineárních bloků a bloků pro zobrazování a ukládání signálů. [3]

## 2.3. Grafické uživatelské rozhraní

V systému Matlab je integrován automatický generátor GUI. Průvodce tvorbou GUI je grafické interaktivní prostředí, které obsahuje všechny grafické objekty typu uicontrol. Vzhled vytvořené aplikace je uložen do souboru s příponou \*.fig a zdrojový kód do souboru s příponou \*.m. Tvorba grafického uživatelského rozhraní by měla sloužit k ulehčení ovládání pro uživatele. Hlavní části grafického interaktivního prostředí je pracovní plocha a paleta komponentů. Paleta komponentů obsahuje základní grafické objekty. Vzhled aplikace se vytváří přesunutím grafických objektů na pracovní plochu. [4]

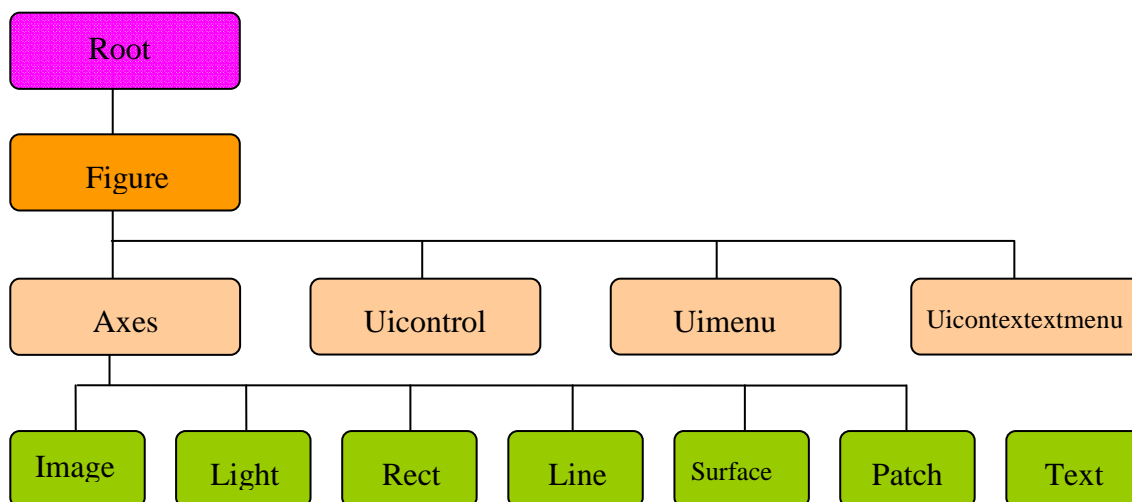
Paleta  
komponentů



Obr. 1 Pracovní okno GUI

## 2.4. Grafické objekty

Grafické objekty jsou všechny prvky určené k zobrazení všech grafických výstupů. U grafických objektů existuje tzv. hierarchie viz obr. 2 , díky níž je možné měnit všechny vlastnosti objektu uicontrol v úrovni Figure. [4]



Obr. 2 Hierarchie grafických objektů

### 3. Teorie analýzy signálů

#### 3.1. Výkonový signál

Výkonový signál  $w(t)$ , je signál, jehož energie  $E$  je rovna nule a střední výkon  $P$  je větší než nula a současně menší než nekonečno.

$$P = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} w^2(t) dt \quad (1)$$

#### 3.2. Energetický signál

Energetický signál  $w(t)$ , je signál, jehož střední výkon  $P$  je roven nule a energie tohoto signálu  $E$  je větší než nula a současně menší než nekonečno.

$$E = \int_{-\infty}^{\infty} w^2(t) dt \quad (2)$$

#### 3.3. Shannon- Kotělníkův teorém

Z Shannon- Kotělníkova teorému vyplývá, že signál lze zcela obnovit ideálním dolnoproputním filtrem, jestliže vzorkovací kmitočet  $\omega_s$  je nejméně dvakrát vyšší než šířka pásma vzorkovaného signálu  $B$ .

$$\omega_s > 2B \quad (3)$$

Jestliže je vzorkovací kmitočet  $\omega_s$  roven šířce pásma  $B$ , mluvíme o Nyquistovým vzorkovacím kmitočtu, někdy také o kritickém úhlovém vzorkování.

$$\omega_s = 2B \quad (4)$$

#### 3.4. Fourierova transformace

Fourierova transformace obecně řadíme mezi tzv. integrální transformace. Slouží pro převod signálů z časové oblasti do oblasti frekvenční. Signál musí být periodický a splňovat Dirichletovy podmínky. Signál může být buď ve spojitém nebo diskrétním čase.

Pomocí Fourierovy transformace se transformuje signál  $w(t)$  na spektrální funkci  $W(\omega)$ . Pomocí zpětné Fourierovy transformace lze transformovat spektrální funkci  $W(\omega)$  na signál  $w(t)$ .

Fourierovou transformací se získává spojitě kmitočtové spektrum. Toto spektrum se využívá pro kmitočtové oblasti signálu, který je fyzikálně realizován s konečnou energií

- Přímá Fourierova transformace

$$W(\omega) = F\{w(t)\} = \int_{-\infty}^{+\infty} w(t) \cdot e^{-j\omega t} dt \quad (5)$$

$$\omega \in (-\infty, +\infty)$$

- Zpětná Fourierova transformace

$$w(t) = F^{-1}\{W(\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{+\infty} W(\omega) \cdot e^{j\omega t} d\omega \quad (6)$$

$$t \in (-\infty, +\infty)$$

### 3.5. Fourierova řada

Fourierova řada je matematickou aproximací vztahu mezi časovou a kmitočtovou oblastí signálu se spojitým časem.

Fourierovou řadou lze získat diskrétní kmitočtové spektrum. Toto spektrum popisuje signál v kmitočtové oblasti diskrétními body.

Pomocí Fourierovy řady se aproximuje signál  $w(t)$  v daném časovém intervalu na koeficienty Fourierovy řady  $w_m$ .

- Přímý výpočet Fourierovy řady

$$w(m) = FR\{w(t)\} = \frac{1}{T} \int_0^T w(t) \cdot e^{-j\omega_m t} dt \quad (7)$$

- Zpětný výpočet Fourierovy řady

$$w(t) = FR^{-1}\{w_m\} = \sum_{m=-\infty}^{m=+\infty} W_m \cdot e^{j\omega_m t} \quad (8)$$

$$\omega_m = m \cdot \omega = m \cdot \frac{2 \cdot \pi}{T}$$

[5], [6]

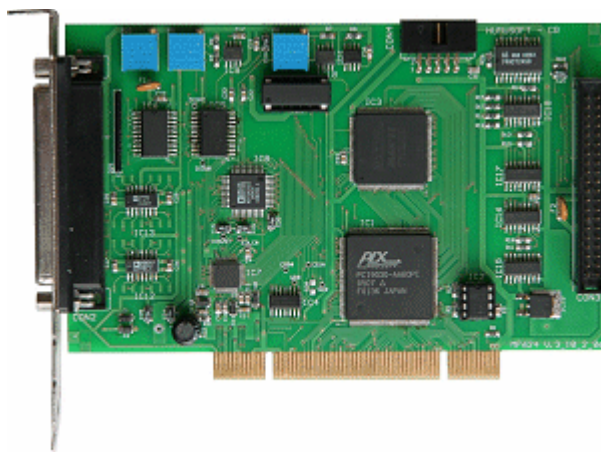
#### 4. Měřicí karta MF 624

Měřicí karta MF 624 je vstupně - výstupní multifunkční karta od firmy Humusoft, určena pro měřicí aplikace. Tato karta obsahuje osm čtrnácti bitových analogových vstupních kanálů, osm čtrnácti bitových analogových výstupních kanálů, osm digitálních vstupů, osm digitálních výstupů a čtyři časovače.

Díky nízké spotřebě energie ji lze využít nejen ve stolních počítačích, ale také v přenosných.

Analogové vstupy a výstupy nabízejí bipolární rozsahy 10V, které vyhovují většině aplikací určených pro měření a regulaci.

Multifunkční karta MF 624 je určená pro sběrnici PCI. Karta má 32bitovou architekturu pro zajištění maximální propustnosti. Ke kartě se dodávají ovladače pro Microsoft Windows a knihovna pro vývoj aplikací. Dále jsou k dispozici i ovladače pro Real Time Toolbox, Real Time Windows Target a xPC. Karty MF 624 jsou vyráběny v souladu se směrnicí RoHS.



Obr. 3 Měřicí karta MF 624

##### 4.1. Technické parametry

- Analogové vstupy

Kanály:	8
A/D převodník:	14.bitový, doba převodu 10 $\mu$ s
Čas převodu	1,6 $\mu$ s 1 kanál
	1,9 $\mu$ s 2 kanály
	2,5 $\mu$ s 4 kanály
	3,7 $\mu$ s 8 kanálů
Vstupní rozsahy	$\pm 10$ V
Výstupní impedance	$10^{10} \Omega$

- Analogové výstupy

- |                 |                          |
|-----------------|--------------------------|
| Kanály:         | 8                        |
| Vstupní rozsah: | $\pm 10V$                |
| Výstupní proud: | maximální proud je 10 mA |
- Digitální vstupy a výstupy
 

Vstupní linky:	8, s úrovněmi TTL
Výstupní linky:	8, s úrovněmi TTL
  - Časovač
 

Kanály:	4
Rozlišení:	32 bit, 20 ns
Módy:	PWM, čítač, generování pulzu
  - Vstupy inkrementálních snímačů
 

Vstupní kanály:	4
Vstupní frekvence:	maximálně 2,5 MHz
Rozlišení:	32 bit
  - Všeobecné údaje
 

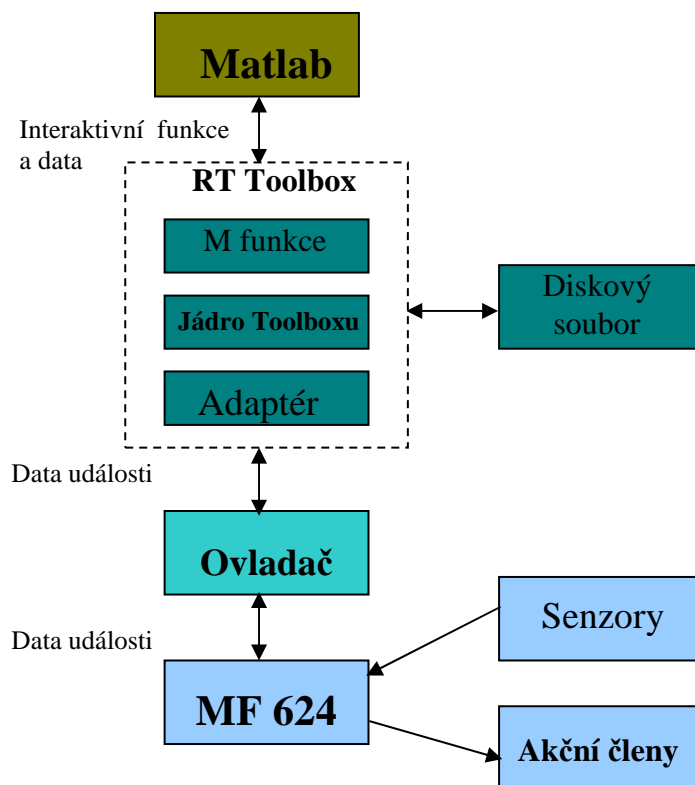
Odběr proudu:	500 mA
Pracovní teplota:	0 – 50 °C
Konektor	2 x DB – 37 F
Sběrnice:	PCI 5V nebo 3,3V

[7], [8]

## 4.2. Real Time Toolbox

Real Time Toolbox umožňuje v prostředí programů Matlab/Simulink práci s externími analogovými a digitálními signály. Je založen na výkonném jádře reálného času. Toto jádro umožňuje vzorkovací frekvence až 66 kHz, bez nutnosti externího časovače. Podporuje širokou škálu měřících karet , včetně multifunkční vstupně výstupní karty MF 624, ale také podporuje sériovou linku RS 232, dále dovede sledovat a zachytávat data z game portu a vyhodnocovat souřadnice z portu myši.

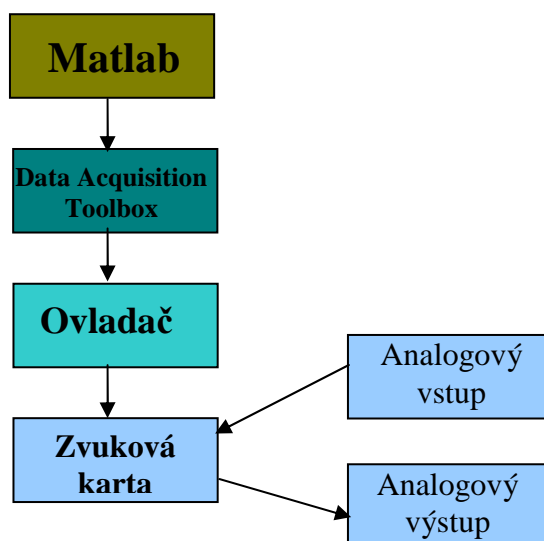
Celý systém, který Matlab využívá pro sběr dat a řízení technologických procesů, znázorňuje blokové schéma na obr. 3. RTT se skládá ze tří částí : M-funkce, jádra toolboxu a adaptéru (ovladače). Vlastní jádro toolboxu plní své úkoly asynchronně s činnostmi Matlabu. [9]



Obr. 4 Struktura systému pro práci Matlabu v reálném čase

## 5. Zvuková karta

ALC 260 je dvoukanálová integrovaná zvuková karta určená především pro stolní počítače a notebooky. Tato karta nabízí dva výstupní kanály a vstupní kanál pro mikrofon, díky čemuž poskytuje kompletní integrované zvukové řešení pro PC. [10]



Obr. 5 Struktura systému pro práci Matlabu se zvukovou kartou



## 6. Realizace výukové aplikace

### 6.1. Struktura aplikace

Původní aplikace byla vytvořena pomocí automatického generátoru GUI. Vzhled vytvořené aplikace je uložen do souboru Aplikace.fig a zdrojový kód do souboru Aplikace.m. Soubor Aplikace.m je hlavní část programu, ve které jsou nastaveny všechny vstupní a výstupní objekty. Vstupní objekty jsou ošetřené proti špatnému zadání hodnoty. Z tohoto souboru se volají funkce, které řeší veškeré výpočty v aplikaci.

Výpis funkcí:

- `fce_energ_harm`  
funkce sloužící pro výpočty a vykreslení energetického harmonického signálu v časové nebo frekvenční oblasti.
- `fce_energ_obdelnik`  
funkce sloužící pro výpočty a vykreslení energetického obdélníkového signálu v časové nebo frekvenční oblasti.
- `fce_energ_pila`  
funkce sloužící pro výpočty a vykreslení energetického pilového signálu v časové nebo frekvenční oblasti.
- `fce_vykon_harm`  
funkce sloužící pro výpočty a vykreslení výkonového harmonického signálu v časové nebo frekvenční oblasti.
- `fce_vykon_obdelnik`  
funkce sloužící pro výpočty a vykreslení výkonového obdélníkového signálu v časové nebo frekvenční oblasti.
- `fce_vykon_pila`  
funkce sloužící pro výpočty a vykreslení výkonového pilového signálu v časové nebo frekvenční oblasti.
- `fce_zvuk_generovat`  
funkce sloužící pro vytvoření vektoru generovaného signálu.
- `fce_zvuk_zobrazit`  
funkce sloužící pro otevření vstupního kanálu zvukové karty a následnému vzorkování vstupního signálu.

Použitím funkcí se aplikace značně zpřehlednila a zjednodušila. Díky tomu byl dosažen menší počet řádků v hlavním souboru Aplikace.m. Nevýhodou funkcí je, že se ve funkci nemůže odkázat na

objekt, který je nadefinovaný v hlavním souboru Aplikace.m. Každá z použitých funkcí má vstupní a výstupní parametry.

Vzhled aplikace je tvořen jedním souborem s názvem Aplikace.fig. To znamená, že obrazovky jsou vytvořené pomocí panelů, u kterých se mění jejich visibility (viditelnost). Obrazovky obsahují vstupní objekty, do kterých jsou zadávány vstupní parametry, axes sloužící k vykreslení proměnných a axes sloužící k vykreslení obrázků. K vykreslení obrázků do axesu slouží příkaz imshow.

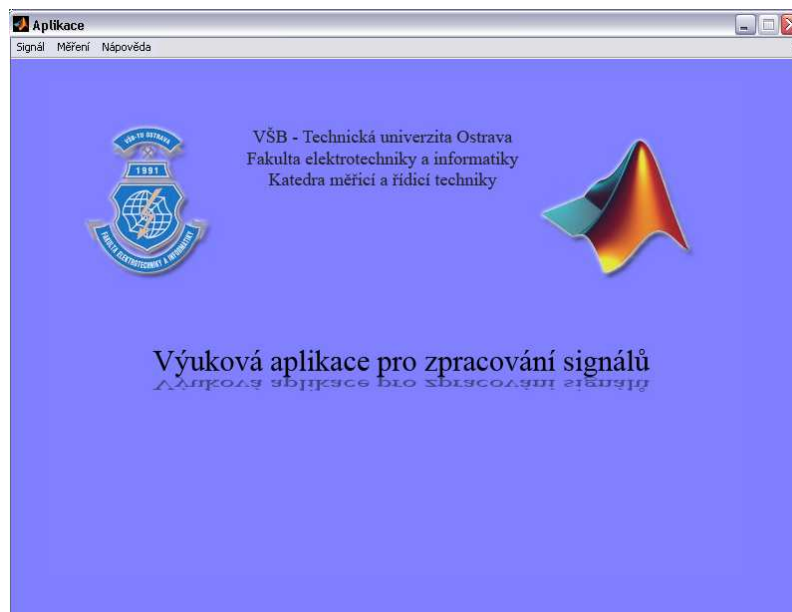
```
function logo_CreateFcn(hObject, eventdata, handles)
axes(hObject);
imshow('logo.jpg');    %% vykreslení obrázku
```

Na obrázku č. 6 je znázorněna hierarchie obrazovek. Z této hierarchie je patrné, že se úvodní obrazovka spustí jen po spuštění aplikace. Po výběru jiné obrazovky se již nelze vrátit k úvodní obrazovce. Mezi ostatními obrazovkami je zpětná vazba.



Obr. 6 Hierarchie obrazovek

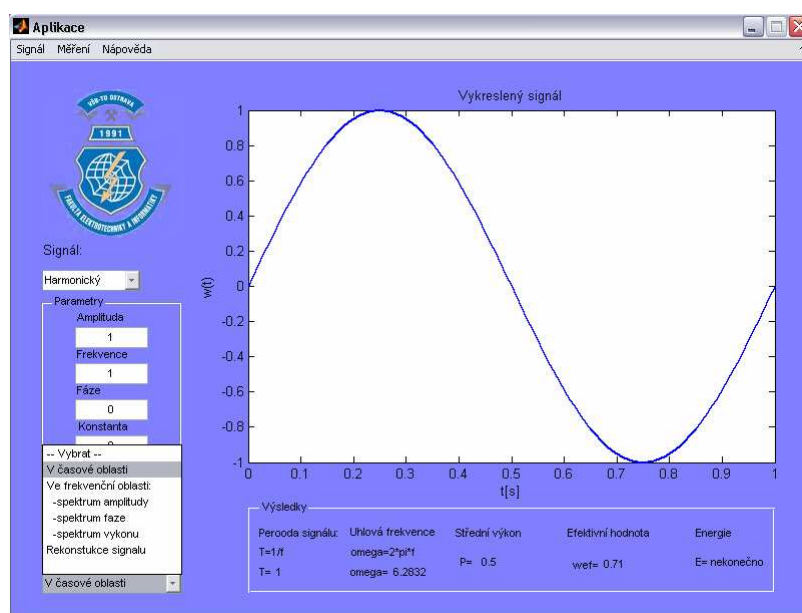
Po spuštění aplikace se zobrazí úvodní obrazovka viz obr.6, ze které je možnost výběru druhu signálů, nápovědy nebo části zabývající se měřením a generováním signálů.



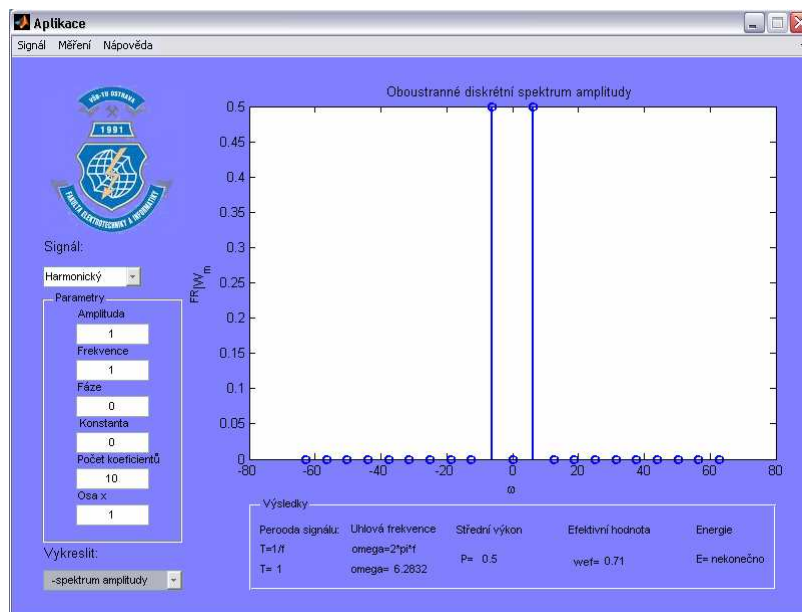
Obr. 7 Úvodní obrazovka

Při zvolení výkonového signálu je na výběr z harmonického, obdélníkového nebo pilového signálu. V panelu „Parametry“ jsou vstupní parametry signálu. Jestliže jsou zadány vstupní parametry nekorektní, aplikace automaticky opraví příslušný parametr. Zadaný signál lze zobrazit v časové nebo frekvenční oblasti. Frekvenční oblast je rozdělena na spektrum amplitudy, fáze a výkonu. Další možností je zobrazení rekonstruovaného signálu. Ze zadaných vstupních parametrů se vypočítá fourierova transformace a pomocí zpětné fourierově transformaci se vykreslí rekonstruovaný signál.

Na obrázku č. 8 je znázorněn výkonový signál v časové oblasti. Po vykreslení signálu jsou zobrazeny parametry signálu jako je střední výkon, efektivní hodnota a energie.

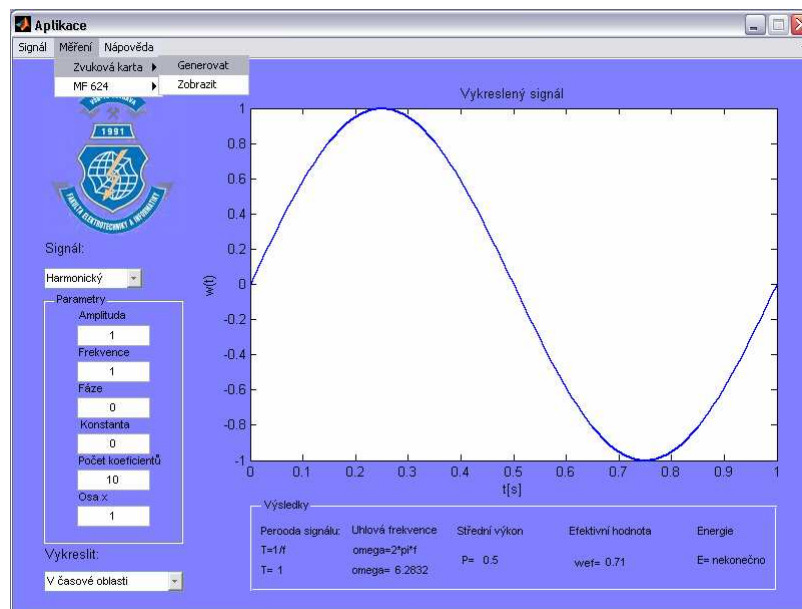


Obr. 8 Vykreslení výkonového signálu v časové oblasti



Obr. 9 Vykreslení výkonového signálu ve frekvenční oblasti

Po vyplnění všech potřebných parametrů lze signál generovat. Generovat lze signál pomocí zvukové karty nebo přídavné karty MF 624. Doba generování signálu je daná velikostí hodnoty vstupního objektu „Osa x“. Na obr. 9 je zobrazen postup generování harmonického signálu pomocí zvukové karty.



Obr. 10 Výstupní obrazovka vyvinuté aplikace pro generování signálu.

Generovat lze jen výkonový signál. Z toho vyplývá, že aplikace musí zjistit jestli je vybrán jeden z výkonových signálů. Toho docílí tak, že zjistí, zda je hodnota popup menu „Signál“ větší než 1 (viz. obr. 10).

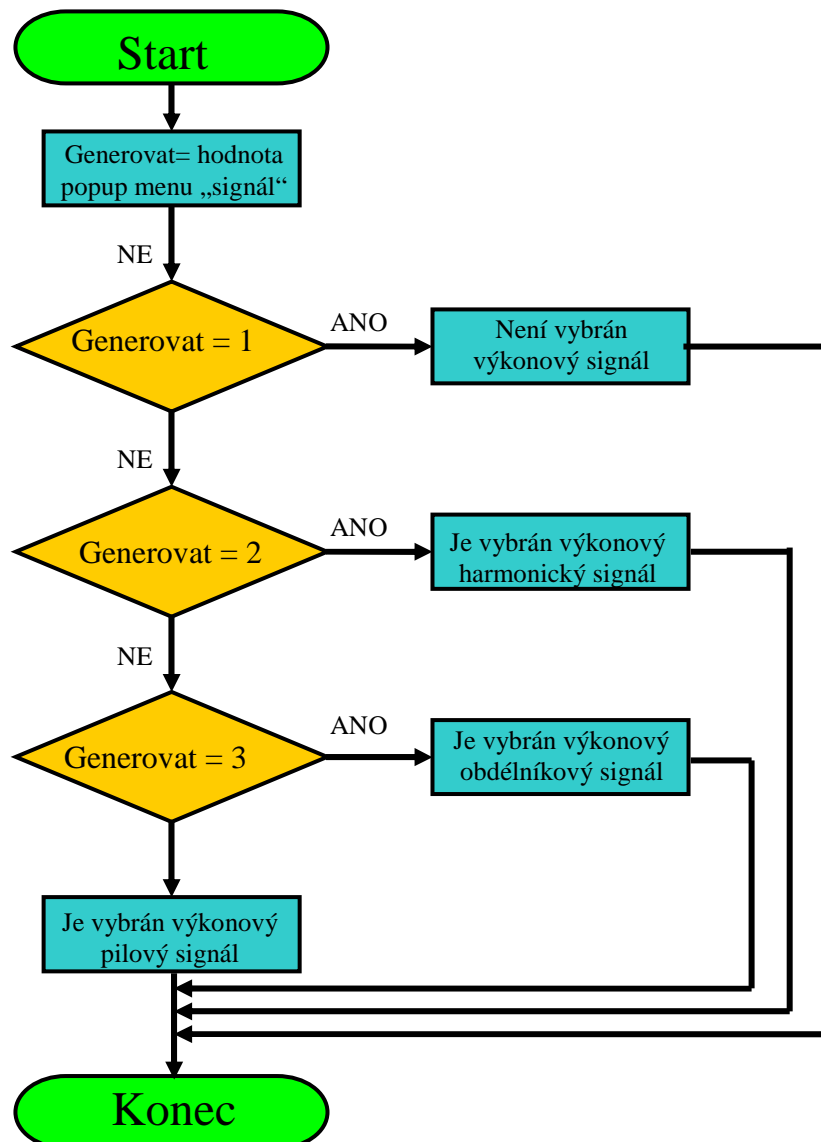
```

generovat = get(handles.popup_vykon_signal,'Value'); %% zjištění hodnoty
popup

if(generovat==1);                                %% není vybrán výkonový signál
    warndlg('Vyberte prosím jeden z výkonových signálů','!! Varování!!!')

else(generovat>1)
    if(generovat==2)                                %% je vybrán harmonický výkonový signál
    elseif(generovat==3)                            % je vybrán obdélníkový výkonový signál
    else(generovat==4)                              % je vybrán pilový výkonový signál
    end
end

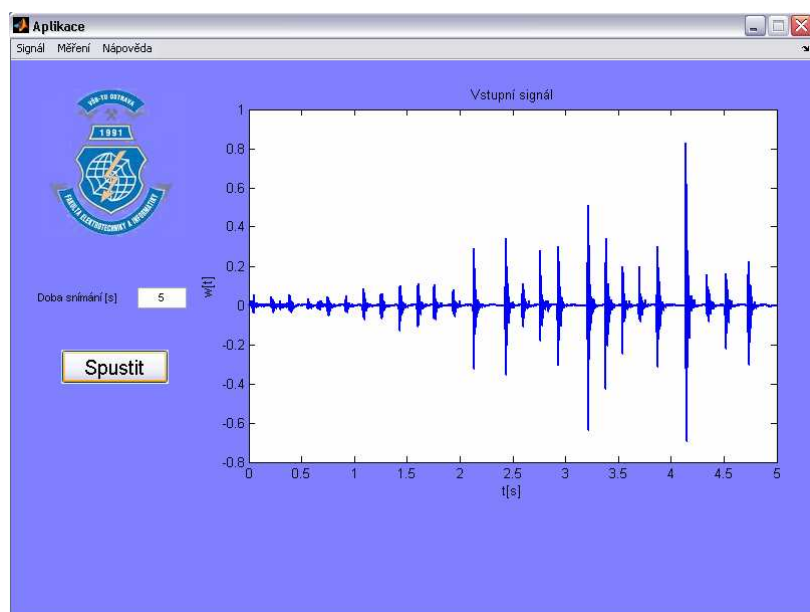
```



Obr. 11 Vývojový diagram – generování

Další možností aplikace je zobrazení signálu z analogového vstupu zvukové karty nebo přídavné karty MF 624. Doba snímání signálu je daná vstupním objektem „Doba snímání“.

Při zobrazování signálu nelze zobrazovat signál v reálném čase. Nejdříve se musí načíst všechny vzorky. Po načtení jsou všechny vzorky zobrazeny ve grafu.



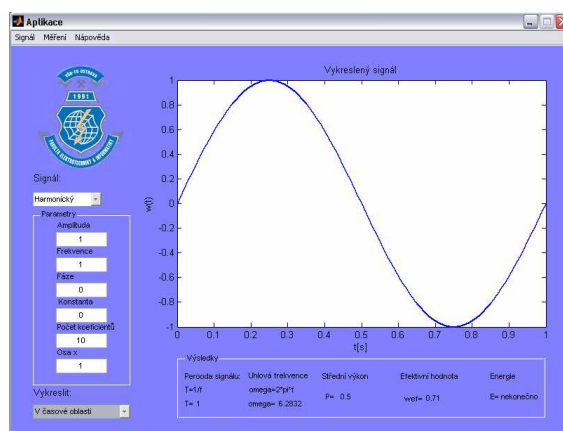
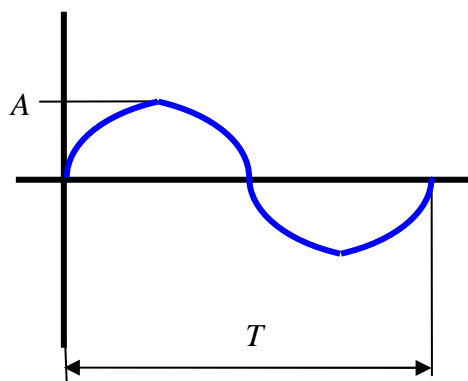
Obr. 12 Zobrazení signálu z analogového vstupu zvukové karty

## 6.2. Modelování signálů

### 6.2.1. Výkonový signál

#### 6.2.1.1. Harmonický signál

Hodnota  $A$  je maximální hodnota signálu zadaná v aplikaci.



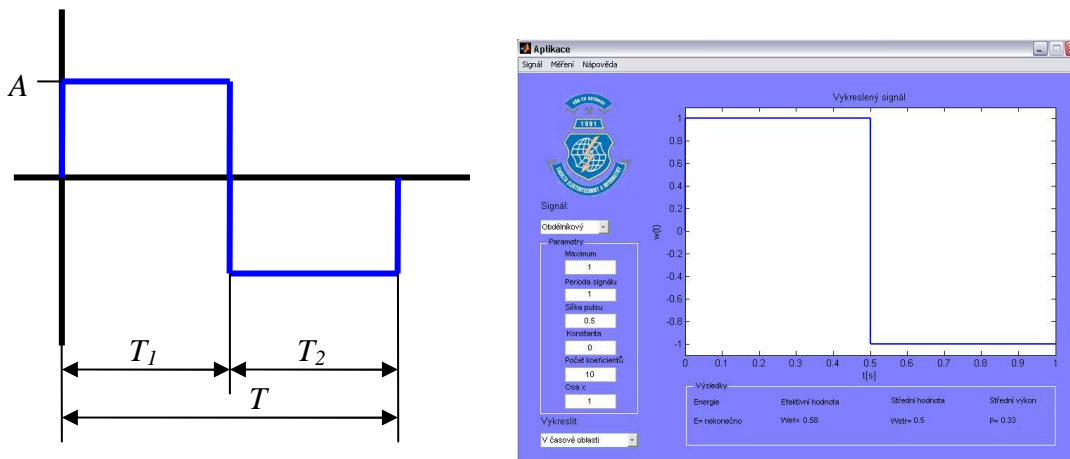
Obr. 13 Způsob modelování výkonového harmonického signálu v aplikaci

Matematický popis harmonického signálu

$$w(t) = A \cdot \sin(2 \cdot \pi \cdot f \cdot t + \varphi) + C$$

### 6.2.1.2. Obdélníkový signál

Obdélníkový signál byl rozdělen na dva časové úseky  $T_1$  a  $T_2$ . Hodnota  $A$  je maximální hodnota signálu zadaná v aplikaci.



Obr. 14 Způsob modelování výkonového obdélníkového signálu v aplikaci

$$T_1 = A$$

$$T_2 = -A$$

Generování výkonového obdélníkového signálu je provedeno následujícím algoritmem:

```
%Generování časového průběhu signálu w(t)
for n=0:VOO:VOO-VOP+2*deltat;

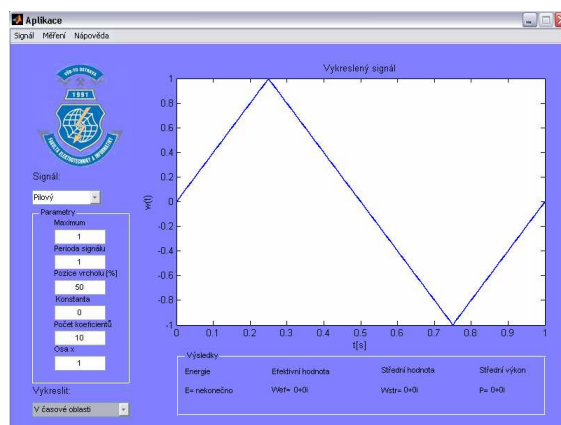
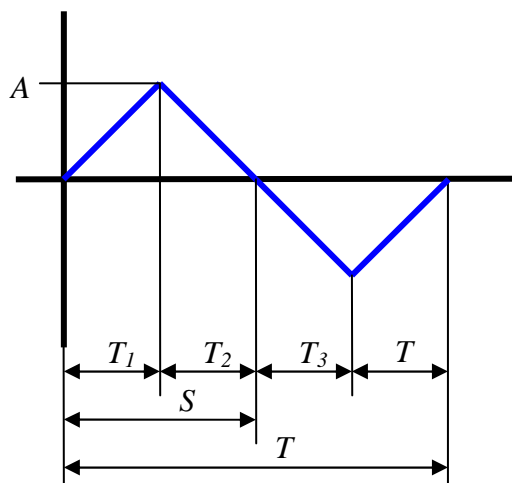
    ta=n:deltat:n+VOS;% 1.část časové osy
    wa(1:length(ta))=VOA+VOK;% 1.část signálu w(t)

    tb=n+VOS:deltat:n+VOP;% 2.část časové osy
    wb(1:length(tb))=-VOA+VOK;% 2.část signálu w(t)

    t=[t ta tb];% spojení časové osy
    w=[w wa wb];% spojení periodického signálu w(t)
end;
```

### 6.2.1.3. Pilový signál

Aby mohl být pilový signál popsán matematickým popisem, byla perioda pilového signálu  $T$  rozdělena na čtyři časové úseky. Rozdělením se získaly čtyři úsečky. Hodnota  $A$  je maximální hodnota signálu zadaná v aplikaci. Tvar signálu je ovlivněn šířkou signálu  $S$ . Ta je v aplikaci zadávána v procentech.



Obr. 15 Způsob modelování výkonového pilového signálu v aplikaci

$$T_1 = \frac{A}{S} \cdot t$$

$$T_2 = \left( \frac{-A}{\frac{T}{2} - S} \right) \cdot \left( t - \frac{T}{2} \right)$$

$$T_3 = \left( \frac{-A}{\frac{T}{2} - S} \right) \cdot \left( t - \frac{T}{2} \right)$$

$$T_4 = \left( \frac{A}{S} \right) \cdot (t - T)$$

Generování výkonového pilového signálu je provedeno následujícím algoritmem:

```
%Generování časového průběhu signálu w(t)
b=2;
a=0;
c=2*T;
d=2*T;
for n=0:2*T:VPO-T+2*deltat;

    ta=n:deltat:n+sirka;
    wa(1:length(ta))=(VPA/sirka)*(ta-a)+VPK;
    tb=n+sirka:deltat:n+T;
    wb(1:length(tb))=(-VPA/(T-sirka))*(tb-(T+a))+VPK;
    tc=n+T:deltat:n+2*T-sirka;
    wc(1:length(tc))=(-VPA/(T-sirka))*(tc-(T+a))+VPK;
    td=n+2*T-sirka:deltat:n+2*T;
    wd(1:length(td))=(VPA/sirka)*(td-c)+VPK;

    t=[t ta tb tc td];
    w=[w wa wb wc wd];
    a=T*b;
    c=c+d;
    b=b+2;

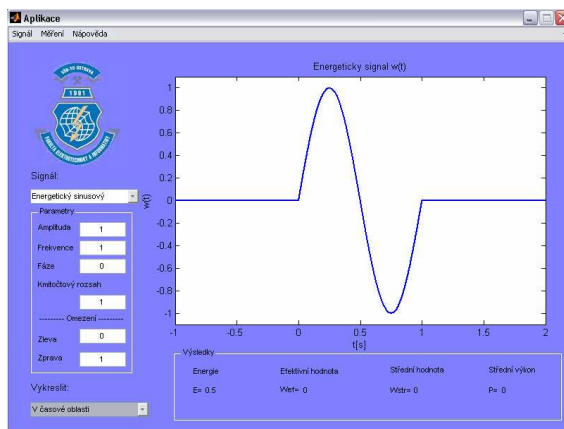
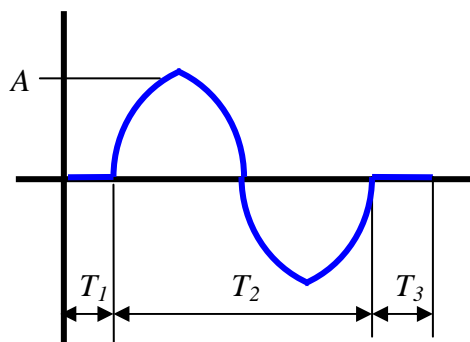
end;
```



## 6.2.2. Energetický signál

### 6.2.2.1. Harmonický signál

Energetický harmonický signál byl rozdělen na tři časové úseky. Signál je omezen v aplikaci vstupními objekty „Zleva“ a „Zprava“



Obr. 16 Způsob modelování energetického harmonického signálu v aplikaci

$$T_1 = 0$$

$$T_2 = A \cdot \sin(2 \cdot \pi \cdot f \cdot t + \varphi)$$

$$T_3 = 0$$

Generování energetického harmonického signálu je provedeno následujícím algoritmem:

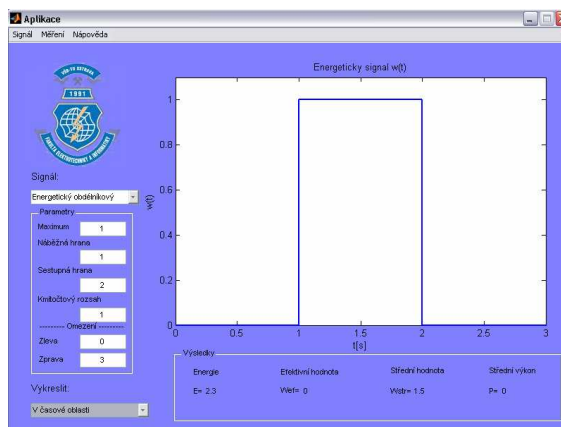
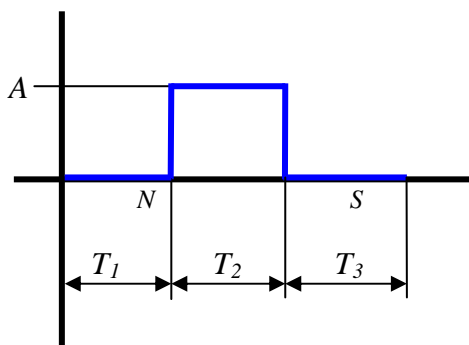
```
%Definice energetického signalu w1(t)
t1a=-1+EHZL:deltat:EHZL-deltat;
w1a(1:length(t1a))=0;

t1b=EHZL:deltat:EHZP;      %cas
w1b(1:length(t1b))=EHA*sin((2*pi*EHF*t1b)+EHP);
t1c=EHZP:deltat:EHZP+1;
w1c(1:length(t1c))=0;

t1=[t1a t1b t1c]; % casova osa
w1=[w1a w1b w1c]; % signal w1(t)
```

### 6.2.2.2. Obdélníkový signál

Energetický obdélníkový signál byl rozdělen na tři časové úseky. Hodnota  $A$  je maximální hodnota signálu zadaná v aplikaci,  $N$  je náběh energetického signálu a  $S$  je sestup signálu.



Obr. 17 Způsob modelování energetického obdélníkového signálu v aplikaci

$$T_1 = 0$$

$$T_3 = 0$$

$$T_2 = A$$

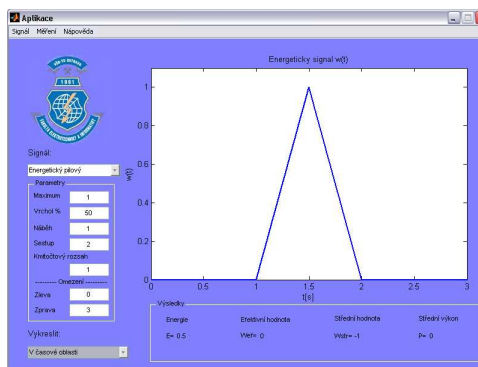
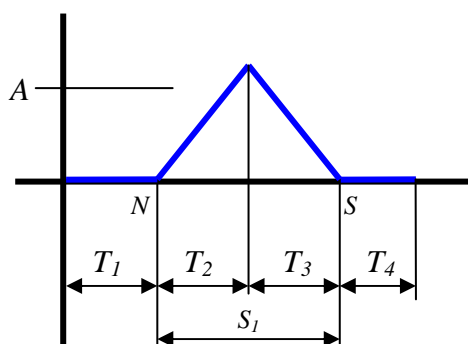
Generování energetického obdélníkového signálu je provedeno následujícím algoritmem:

```
t1a=(EOZL:deltat:EON);
w1a(1:length(t1a))=0;
t1b=EON:deltat:EOS;
w1b(1:length(t1b))=EOM;
t1c=EOS:deltat:EOZP;
w1c(1:length(t1c))=0;
```

```
t1=[t1a t1b t1c];
w1=[w1a w1b w1c];
```

### 6.2.2.3. Pilový signál

Energetický pilový signál byl rozdělen na čtyři časové úseky. Hodnota  $A$  je maximální hodnota signálu zadaná v aplikaci,  $N$  je náběh energetického signálu a  $S$  je sestup signálu. Tvar signálu je ovlivněn šířkou signálu  $S_1$ . Ta je v aplikaci zadávána v procentech.



Obr. 18 Způsob modelování energetického pilového signálu v aplikaci

$$\begin{aligned}
T_1 &= 0 \\
T_2 &= \left( \frac{A}{S_1} \right) \cdot (t - N) \\
T_3 &= \left( -\frac{A}{S_1} \right) \cdot (t - S) \\
T_4 &= 0
\end{aligned}$$

Generování výkonového pilového signálu je provedeno následujícím algoritmem:

```

t1a=(EPZL:deltat:EPN);
w1a(1:length(t1a))=0;

t1b=EPN:deltat:EP_P;
w1b(1:length(t1b))=(EPM/Ep2)*(t1b-EPN);

t1c=EP_P:deltat:EPS;
w1c(1:length(t1c))=(-EPM/Ep2)*(t1c-EPS);

t1d=EPS:deltat:EPZP;
w1d(1:length(t1d))=0;

vystupni2=[t1a t1b t1c t1d];
vystupni=[w1a w1b w1c w1d];

```

### 6.3. Generování a měření signálů zvukovou kartou

#### 6.3.1. Generování signálu

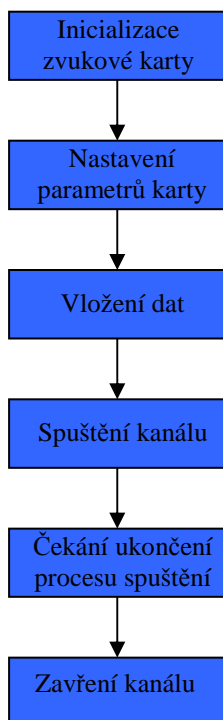
Při práci s analogovým výstupem se nejprve musí vytvořit objekt zařízení s přiřazením jednotlivých kanálů. Pro uložení dat do jednotky zpracování, slouží funkce putdata, která blokuje funkci Matlabu, až po dobu, než dojde k uložení všech vzorků. Po uložení dat pro vytvoření analogového výstupního signálu je možné spustit generování zvukovou kartou funkcí start. Funkce wait slouží k čekání na ukončení práce zařízení. Po skončení generování signálu se odstraní objekt zařízení.

Přehled funkcí a vlastností:

Analoginput	vytvoří objekt pro analogový vstup
Analogoutput	vytvoří objekt pro analogový vstup
Addchanel	přiřadí kanál k objektu analogového vstupu nebo výstupu
Start	spustí objekt zařízení
Stop	zastaví objekt zařízení
Wait	čeká až skončí činnost zařízení
Peekdata	náhled naposledy zaznamenaných vzorků
Putdata	zapiše hodnoty do fronty analogového výstupu
Getdata	vyčte vzorky z jednotky zpracování dat
Clear	odstraní objekt z pracovního prostoru Matlabu
Close	zavře objekt zařízení
Delete	odstraní objekty zařízení
SampleRate	vzorkovací frekvence
SamplesPerTrigger	počet vzorků pro každý vstup při jednom spuštění
SamplesAcquired	okamžitý počet vstupních vzorků

RepeatOutput            opakuje celý blok dat

```
ao= analogoutput('winsound');    %%vytvoření objektu pro analogový  
                                      výstup  
kanal=addchannel(ao,1);            %%přiřazení kanálu k Objektu  
                                      analogového výstupu  
set(ao,'SampleRate',10000)        %%nastavení vzorkovací frekvence  
set(ao,'RepeatOutput',0);        %%nastavení opakování  
putdata(ao,s1);                    %%zapiše hodnoty do fronty  
start(ao)                          %%spustí objekt zařízení  
wait(ao,preruseni)                %%čeká až skončí činnost zařízení  
stop(ao)                            %%zastaví objekt zařízení  
delete(ao);                        %%odstraní objekt zařízení  
clear ao;
```



Obr. 19 Princip generování signálu zvukovou kartou

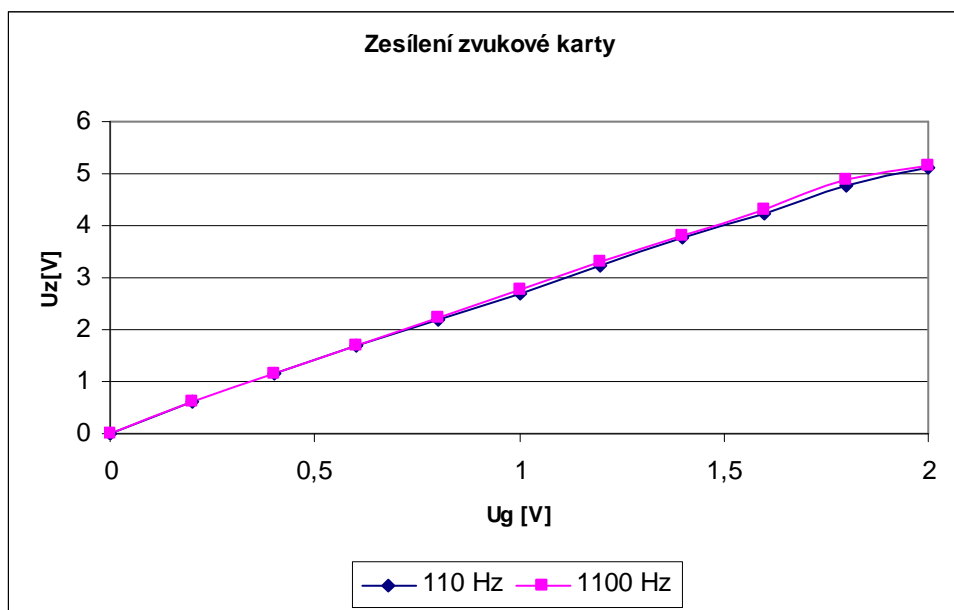
Měřením bylo zjištěno, že generovaný signál byl zesílený oproti signálu, který byl vytvořen pomocí Matlabu. Je to způsobené vnitřním zesilovačem zvukové karty.

Při měření zesílení zvukové karty bylo softwarově nastaveno nejvyšší výstupní zesílení karty. Signál, generovaný zvukovou kartou měl harmonický průběh. Měření bylo uskutečněno pro kmitočty 110 Hz a 1100 Hz. Při měření byla zvyšována maximální hodnota napětí generovaného signálu  $U_g$  do doby, než došlo k saturaci. Na výstupu zvukové karty byla měřena hodnota  $U_z$ . Uvedené hodnoty napětí jsou hodnoty špička-špička. K saturaci došlo při nastavení hodnoty napětí generovaného signálu na

2,1V. Z toho lze usoudit, že maximální napětí, které dokáže zvuková karta generovat je 5,2 V špička-špička. Odchylka v tabulce hodnot je rozdíl výstupního signálu  $U_z$  k hodnotě zadávaného signálu  $U_a$  v aplikaci.

Zadávaný signál $U_a$	Generovaný signál $U_g$	110 Hz				1100Hz			
		Výstupní signál $U_z$	Zesílení A	Odchylka $U_o$	Odchylka $U_{o\%}$	Výstupní signál $U_z$	Zesílení A	Odchylka $U_o$	Odchylka $U_{o\%}$
[V]	[V]	[V]	[V]	[V]	[%]	[V]	[V]	[V]	[%]
0,54	0,20	0,62	3,08	0,08	13,89	0,62	3,12	0,08	15,56
1,08	0,40	1,15	2,88	0,07	6,76	1,17	2,92	0,09	8,24
1,62	0,60	1,68	2,80	0,06	3,77	1,68	2,80	0,06	3,70
2,16	0,80	2,19	2,74	0,03	1,44	2,22	2,78	0,06	2,78
2,70	1,00	2,70	2,70	0,00	0,04	2,76	2,76	0,06	2,22
3,24	1,20	3,24	2,70	0,00	0,03	3,30	2,75	0,06	1,85
3,78	1,40	3,78	2,70	0,00	0,00	3,80	2,71	0,02	0,53
4,32	1,60	4,24	2,65	-0,08	-1,85	4,32	2,70	0,00	0,00
4,86	1,80	4,76	2,65	-0,10	-2,04	4,88	2,71	0,02	0,41
5,40	2,00	5,12	2,56	-0,28	-5,19	5,16	2,58	-0,24	-4,44

Tab.1 Tabulka hodnot pro určení zesílení zvukové karty



Obr. 20 Graf znázorňující zesílení zvukové karty

Z grafu naměřených hodnot lze vyčíst, že zesílení zesilovače zvukové karty je konstantní. Změna kmitočtu generovaného signálu nemá vliv na velikost zesílení. Tímto měřením mohla být provedena úprava maximální hodnoty generovaného signálu tak, aby maximální hodnota zadaná v aplikaci byla shodná s hodnotou generovaného signálu na výstupu zvukové karty.

Výpočet zesílení:

Hodnoty uvedené ve výpočtu se vztahují k třetímu řádku tabulky č. 1 pro frekvenci 110Hz.

$$A = \frac{U_z}{U_g} \quad (9)$$

tedy

$$A = \frac{1,15}{0,4}$$

$$A = 2,88$$

Dosažením do vzorce č. 9 vyšlo zesílení zvukové karty 2,88.

Pro větší přesnost byla vypočtena průměrná hodnota zesílení  $A_1$ . Při počítání průměrného zesílení byla odstraněna nejvyšší a nejnižší hodnota zesílení, aby se eliminovala chyba měření. Hodnoty uvedené ve výpočtu průměrného zesílení se vztahují k frekvenci 110Hz tabulky č. 1, kde  $n$  je počet hodnot zesílení po odstranění nejvyšší a nejnižší hodnoty.

$$A_1 = \frac{\sum U_z}{n} \quad (10)$$

tedy

$$A_1 = \frac{2,88 + 2,80 + 2,74 + 2,70 + 2,70 + 2,70 + 2,65 + 2,65}{8}$$

$$A_1 = \frac{21,82}{8}$$

$$A_1 = 2,7$$

Nyní se upraví maximální hodnota generovaného signálu. Tím je dosaženo, že zvuková karta bude generovat signál o stejné maximální hodnotě jako je maximální hodnota zadaná v aplikaci.

`A=VHA/2.7; %%úprava amplitudy podle zesílení`

Aby nedošlo k saturaci vlivem generování signálu o větší maximální hodnotě než je zvuková karta schopna generovat, byla aplikace upravena tak, že se maximální hodnota nastaví na hodnotu dva. Je to maximální hodnota, kterou zvládne zvuková karta generovat, aniž by došlo k saturaci.

```
if (VHA>2) %% Zamezení saturace
    VHA=2; %% Nastavení maximální hodnoty
end
```

Výpočet odchylky:

Hodnoty uvedené ve výpočtu se vztahují k třetímu řádku tabulky č. 1 pro frekvenci 110Hz.

$$U_o = U_z - U_a \quad (11)$$

tedy

$$U_o = 1,68 - 1,62$$

$$U_o = 0,06V$$

$$U_{o\%} = \frac{U_o}{U_a} \cdot 100 \quad (12)$$

tedy

$$U_{o\%} = \frac{0,06}{1,62} \cdot 100$$

$$U_{o\%} = 3,77\%$$

Měření frekvence generovaného signálu:

Měřením bylo zjištěno, že Matlab není schopen nastavit libovolnou vzorkovací frekvenci zvukové karty. Jakmile by se nastavila vzorkovací frekvence, která není 8000, 11 025, 16 000, 22 025 nebo 44 100 Hz, Matlab sám nastaví nejbližší z těchto frekvencí.

Změnou vzorkovací frekvence zvukové karty Matlabem by došlo k nesprávnému vzorkování generovaného signálu, jelikož by nebyla shodná s vzorkovací frekvencí zvukové karty. Tímto by byl generován signál s jinou frekvencí, než byla frekvence zadaná v aplikaci.

Z tohoto důvodu byla nastavená vzorkovací frekvence zvukové karty v aplikaci na 8kHz, aby se zamezilo nesprávnému vzorkování signálu.

Z toho vyplývá, že frekvence signálu zadaného v aplikaci nesmí být větší než 4kHz, jinak by nebyl splněn Shannon- Kotělníkův teorém.

```
if (VOF>4000)    %%Ošetření Shannon- Kotělníkova teorému
    VOF=4000;    %%Nastavení frekvence signálu
end
```

Aby mohla být nastavovaná libovolná vzorkovací frekvence zvukové karty, muselo by dojít k přepočtu frekvence generovaného signálu. Tento přepočet by se musel provést pro každou variantu vzorkovací frekvence zvukové karty.

Pro demonstrativní přepočet frekvence generovaného signálu  $f_g$  bylo provedeno měření s vzorkovací frekvencí signálu a vzorkovací frekvencí karty o hodnotě 10 kHz. Na výstupu zvukové karty byla měřena frekvence výstupního signálu  $f_z$ .

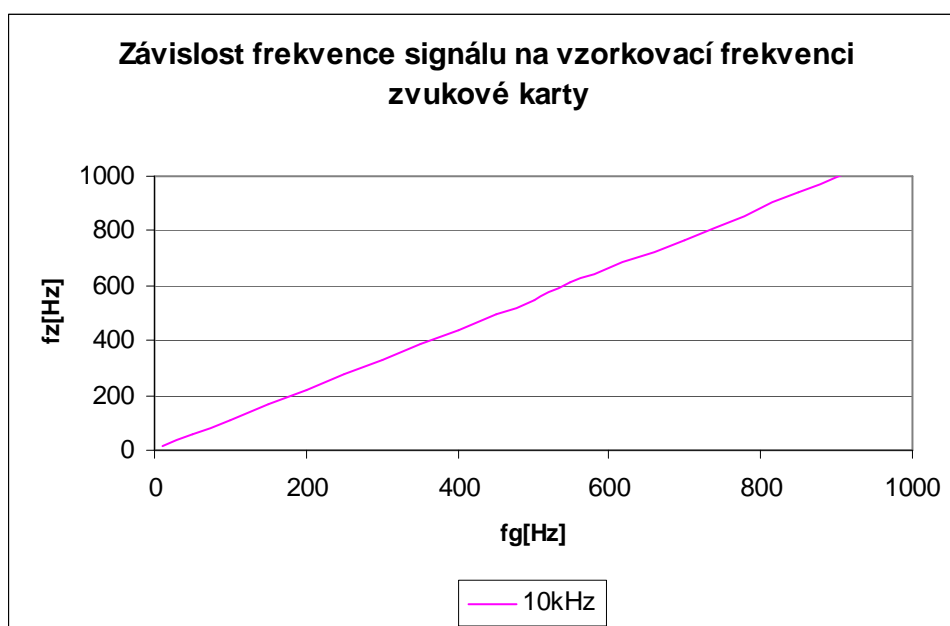
Pomocí příkazu:

```
F_vz=get(ao,'SampleRate');
```

bylo zjištěno, že Matlab nastavil vzorkovací frekvenci zvukové karty na 11 025 Hz. Generovaný signál zadáný v aplikaci byl vzorkován frekvencí 10 000 Hz. Nesprávným vzorkováním byl generován signál o jiné frekvenci než byla hodnota zadaná v aplikaci.

fg	fz
f[Hz]	f[Hz]
10	11,01
50	55
100	110
150	165
200	220
250	276
300	331
350	385
400	440
450	500
500	550
550	610
700	770
850	943
950	1042

Tab.2 Tabulka hodnot pro úpravu frekvence



Obr. 21 Graf závislosti frekvence zvukové karty



Z grafu naměřených hodnot lze vyčíst, že změřená frekvence na výstupu karty lineárně roste s rostoucí frekvencí generovaného signálu.

Z tabulky č. 2 lze je patrné, že změřená hodnota  $f_z$  je o 10% větší než generovaná frekvence  $f_g$ .

Přepočet frekvence by byl proveden podle následujícího vztahu:

$$f = f_g - \left( \frac{f_g}{10} \right) \quad (13)$$

Při zadání frekvence signálu  $f = 100\text{Hz}$  v aplikaci by došlo k následnému přepočtu.

$$f = 100 - \left( \frac{100}{10} \right)$$

$$f = 90\text{Hz}$$

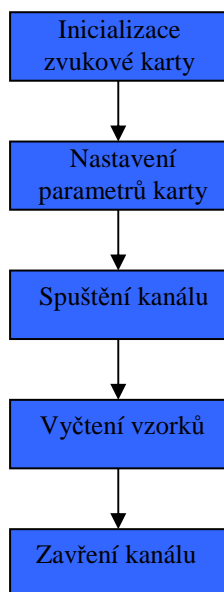
Generovaný signál by měl frekvencí 90Hz. Díky přepočtu by byl na výstupu zvukové karty změřen signál o frekvenci  $f_z = 100\text{Hz}$ .

### 6.3.2. Měření a zobrazení signálu

Při práci s analogovým vstupem se nejprve musí vytvořit objekt zařízení. Poté se přiřadí kanál, na který bude signál přiváděn. Spuštěním objektu zařízení se vyčtou vzorky z jednotky zpracování dat. Po skončení vzorkování signálu se odstraní objekt zařízení. Vyčtené vzorky se zobrazí v grafu.

Vstupní kanály jsou vzorkovány frekvencí 8kHz. Tato frekvence se používá v telekomunikační technice, kde je kladen důraz na srozumitelnost.

```
ai=analoginput('winsound');           %%vytvoření objektu pro analogový vstup
addchannel(ai,1);                      %%přiřazení kanálu k Objektu
interval=2;
vzorkovani=get(ai,'SampleRate');       %%načtení hodnoty vzorkovací frekvence
trigger=interval*vzorkovani
set(ai,'SamplesPerTrigger',trigger),   %%počet vzorků při jednom spuštění
start(ai)                             %%spustí objekt zařízení
vzorky=getdata(ai);                   %%vyčte vzorky z jednotky zpracování dat
delete(ai);                           %%odstraní objekt zařízení
clear ai
```



Obr. 22 Princip měření signálu zvukovou kartou

Pro ověření funkčnosti zobrazení signálu pomocí zvukové karty bylo softwarově nastaveno nejvyšší vstupní zesílení karty. Generovaný signál generátorem byl harmonického průběhu. Měření bylo uskutečněno pro kmitočty 110 Hz a 1100 Hz. Při měření byla zvyšována maximální hodnota napětí  $U_g$  generovaného signálu. V aplikaci byla měřena hodnota  $U_z$ . Uvedené hodnoty napětí jsou hodnoty špička-špička.

	110Hz		1100Hz	
Generovaný signál $U_g$	Změřený signál $U_z$	Zesílení A	Změřený signál $U_z$	Zesílení A
[mV]	[mV]		[mV]	
20,00	220,00	11,00	200,00	10,00
30,00	324,00	10,80	340,00	11,33
40,00	460,00	11,50	420,00	10,50
50,00	506,00	10,12	534,00	10,68
60,00	650,00	10,83	620,00	10,33
70,00	660,00	9,43	760,00	10,86
80,00	810,00	10,13	820,00	10,25
90,00	960,00	10,67	1000,00	11,11
100,00	1060,00	10,60	1100,00	11,00
110,00	1100,00	10,00	1200,00	10,91
120,00	1200,00	10,00	1240,00	10,33

Tab.3 Tabulka hodnot pro měření analogového vstupu zvukové karty

Příklad výpočtů zesílení je dán dosazením hodnot z druhého řádku tabulky č. 3 s frekvencí 110Hz do vzorce č. 9.

$$A = \frac{324}{30}$$

$$A = 10,80$$

Při zobrazování signálu zvukovou kartou je maximální hodnota vstupního signálu přiváděného na vstup zvukové karty značně omezena. Bylo změřeno, že při přivedení signálu o větší hodnotě jak 120mV špička- špička na vstup zvukové karty dochází k saturaci. Je to dáno tím, že zvuková karta používá jako vstupní zařízení mikrofon. Měřené napětí v aplikaci bylo desetkrát vyšší, než bylo generované napětí generátorem.

## 6.4. Generování a měření signálů měřicí kartou MF624

### 6.4.1. Generování signálu

Real Time Toolbox nabízí v Simulinku funkční bloky určené pro práci s měřicími kartami. Jednou z podporovaných karet je i MF 624. Model určený pro analogový výstup je nejprve načten a pomocí příkazů v m-filu jsou měněny parametry jednotlivých funkčních bloků. Velikost vstupního napětí přiváděného signálu je omezena rozsahem  $\pm 10V$ .

Přehled funkcí a vlastností:

load_system	načte model Simulinku do paměti, aniž by bylo okno modelu viditelné
get_param	získá vlastnosti funkčního bloku
set_param	nastaví vlastnosti funkčního bloku
sim	simulace modelu
DialogParameters	získá parametry o funkčním bloku

```
%% Načtení modelu
load_system('pokus_RT.mdl');

%% získání parametru bloku Generátor
p = get_param('pokus_RT/Generator', 'DialogParameters')

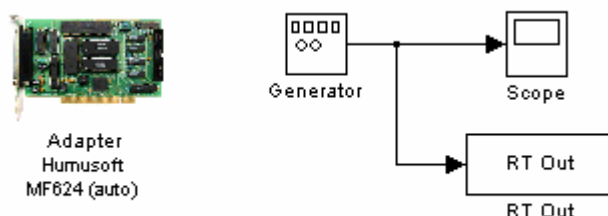
%% nastavení amplitudy generátoru
set_param('pokus_RT/Generator', 'Amplitude', A)

%% nastavení frekvence generátoru
set_param('pokus_RT/Generator', 'Frequency', f)

%% nastavení průběhu generátoru
set_param('pokus_RT/Generator', 'Waveform', 'square')

%% spuštění simulace
sim('pokus_RT', simulace)
```

V modelu pro generování signálu je zapojen blok Signal generátor (Generátor), jehož výhodou je možnost generování harmonického, obdélníkového a pilového signálu. Vygenerovaný signál je přiveden na první kanál analogového výstupu karty.



Obr. 23 Schéma zapojení analogového výstupu měřicí karty v Simulinku

Pro ověření funkčnosti generování signálu pomocí měřicí karty MF 624 bylo provedeno měření. Generovaný signál měl harmonický průběh, jehož parametry se zadávaly v aplikaci. Měření se uskutečnilo pro frekvence generovaného signálu 110 a 1100 Hz. V aplikaci se nastavila amplituda signálu  $U_a$  a na výstupu karty se měřila hodnota generovaného signálu  $U_z$

	110Hz			1100Hz		
Zadávaný signál $U_a$	Změřený signál $U_z$	Odchylka $U_o$	Odchylka $U_{o\%}$	Změřený signál $U_z$	Odchylka $U_o$	Odchylka $U_{o\%}$
[V]	[V]	[V]	[%]	[V]	[V]	[%]
1,00	1,12	0,12	12,00	1,10	0,10	10,00
1,50	1,60	0,10	6,67	1,62	0,12	8,00
2,00	2,08	0,08	4,00	2,10	0,10	5,00
2,50	2,64	0,14	5,60	2,64	0,14	5,60
3,00	3,16	0,16	5,33	3,28	0,28	9,33
3,50	3,72	0,22	6,29	3,76	0,26	7,43
4,00	4,20	0,20	5,00	4,10	0,10	2,50
4,50	4,72	0,22	4,89	4,80	0,30	6,67
5,00	5,24	0,24	4,80	5,36	0,36	7,20
5,50	5,76	0,26	4,73	5,84	0,34	6,18
6,00	6,24	0,24	4,00	6,16	0,16	2,67
6,50	6,76	0,26	4,00	6,64	0,14	2,15
7,00	7,24	0,24	3,43	7,44	0,44	6,29
7,50	7,70	0,20	2,67	8,00	0,50	6,67
8,00	8,40	0,40	5,00	8,40	0,40	5,00

Tab.3 Tabulka hodnot pro měření analogového výstupu měřicí karty

Z tabulky č. 3 je patrné, že velikost amplitudy signálu nastavené v aplikaci byla shodná s velikostí amplitudy na výstupu karty. Odchyly mohly být způsobeny nepřesností měřicího osciloskopu nebo šumem.

Příklady výpočtů odchylek jsou dány pro hodnoty druhého řádku s frekvencí 110Hz.

$$U_o = U_z - U_a \quad (14)$$

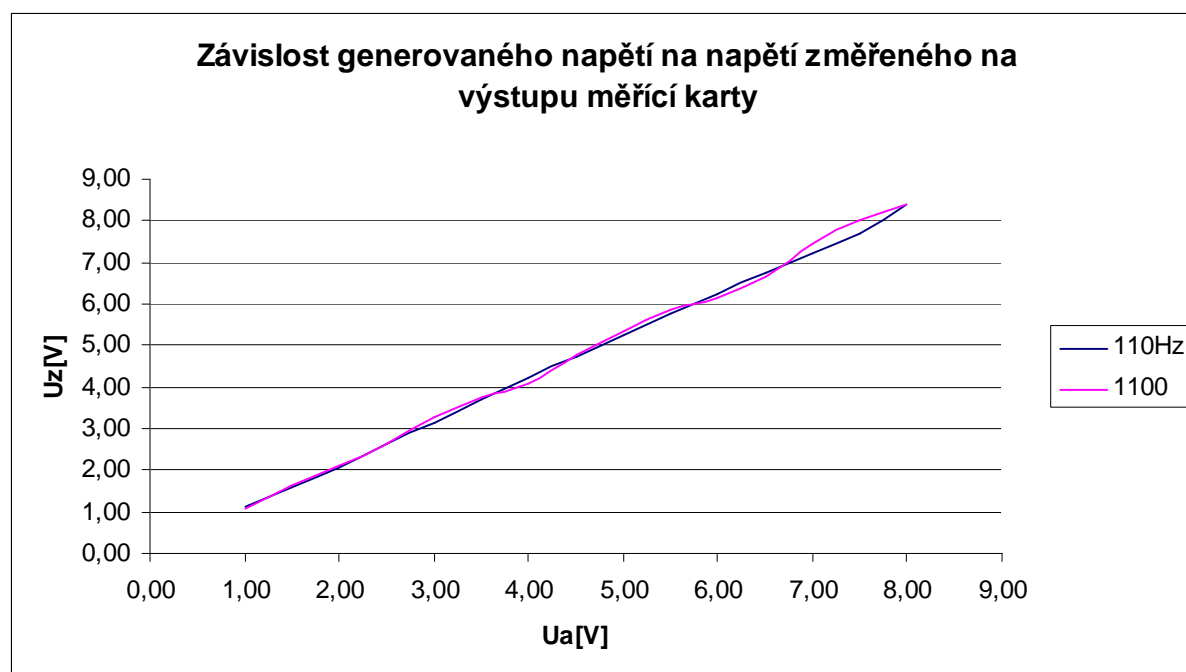
$$U_o = 1,60 - 1,50$$

$$U_o = 0,10V$$

$$U_{o\%} = \frac{U_o}{U_a} \cdot 100 \quad (15)$$

$$U_{o\%} = \frac{0,10}{1,50} \cdot 100$$

$$U_{o\%} = 6,67\%$$



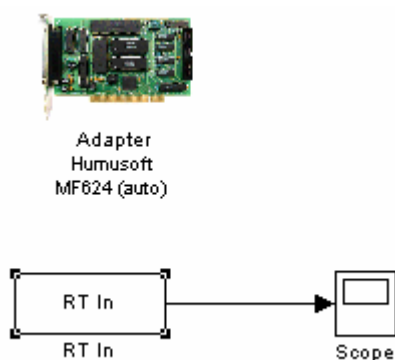
Obr. 24 Graf závislosti generovaného napětí aplikací na napětí změřeného na výstupu měřicí karty

#### 6.4.2. Zobrazení signálu

Zobrazení signálu v aplikaci je řešeno načtením modelu vytvořeného v Simulinku a načítáním vstupních dat pomocí m-filu. Na vstup karty může být přiveden signál o maximální hodnotě  $\pm 10V$ . Překročením této hodnoty může dojít k nevratnému zničení vstupního kanálu. Vzorkování měřicí karty

bylo nastaveno na 20kHz. Omezení vzorkovací frekvence bylo dáno použitím operačního systému Windows Xp od společnosti Microsoft.

```
load_system('RT_in.mdl');           %%načtení modelu Simulinku
sim('RT_in',interval)               %%spuštění simulace
```



Obr. 25 Schéma zapojení analogového vstupu měřicí karty v Simulinku

Pro ověření funkčnosti analogového vstupu měřicí karty MF 624 bylo provedeno měření. Generovaný signál měl harmonický průběh, jehož parametry se zadávaly na měřícím generátoru. Měření se uskutečnilo pro frekvence generovaného signálu 110 a 1100 Hz. Na generátoru se nastavila amplituda signálu  $U_g$  a pomocí karty aplikace se měřila hodnota generovaného signálu  $U_z$ .

	110Hz			1100Hz		
Generovaný signál $U_g$	Změřený signál $U_z$	Odchylka $U_o$	Odchylka $U_{o\%}$	Změřený signál $U_z$	Odchylka $U_{o\%}$	Odchylka $U_{o\%}$
[V]	[V]	[V]	[%]	[V]	[V]	[%]
1,00	0,92	-0,08	-8,00	0,90	-0,10	-10,00
1,50	1,56	0,06	4,00	1,43	-0,07	-4,67
2,00	1,90	-0,10	-5,00	1,94	-0,06	-3,00
2,50	2,40	-0,10	-4,00	2,44	-0,06	-2,40
3,00	2,98	-0,02	-0,67	2,89	-0,11	-3,60
3,50	3,32	-0,18	-5,14	3,40	-0,10	-2,86
4,00	3,78	-0,22	-5,50	3,96	-0,04	-1,00
4,50	4,30	-0,20	-4,44	4,36	-0,14	-3,11
5,00	4,76	-0,24	-4,80	4,95	-0,05	-1,00
5,50	5,30	-0,20	-3,64	5,36	-0,14	-2,55
6,00	5,64	-0,36	-6,00	5,84	-0,16	-2,67
6,50	6,24	-0,26	-4,00	6,36	-0,14	-2,15
7,00	6,70	-0,30	-4,29	6,64	-0,36	-5,14
7,50	7,20	-0,30	-4,00	7,06	-0,44	-5,87
8,00	7,60	-0,40	-5,00	Omezení generátoru		

Tab.4 Tabulka hodnot pro měření analogového vstupu měřicí karty

Z tabulky č. 4 lze vyčíst, že napětí generovaného signálu bylo téměř totožné se změřeným napětím v aplikaci. Při frekvenci 1100Hz generovaného signálu nebylo možné nastavit amplitudu na hodnotu 8V. Je to dáno malým výkonem generátoru.

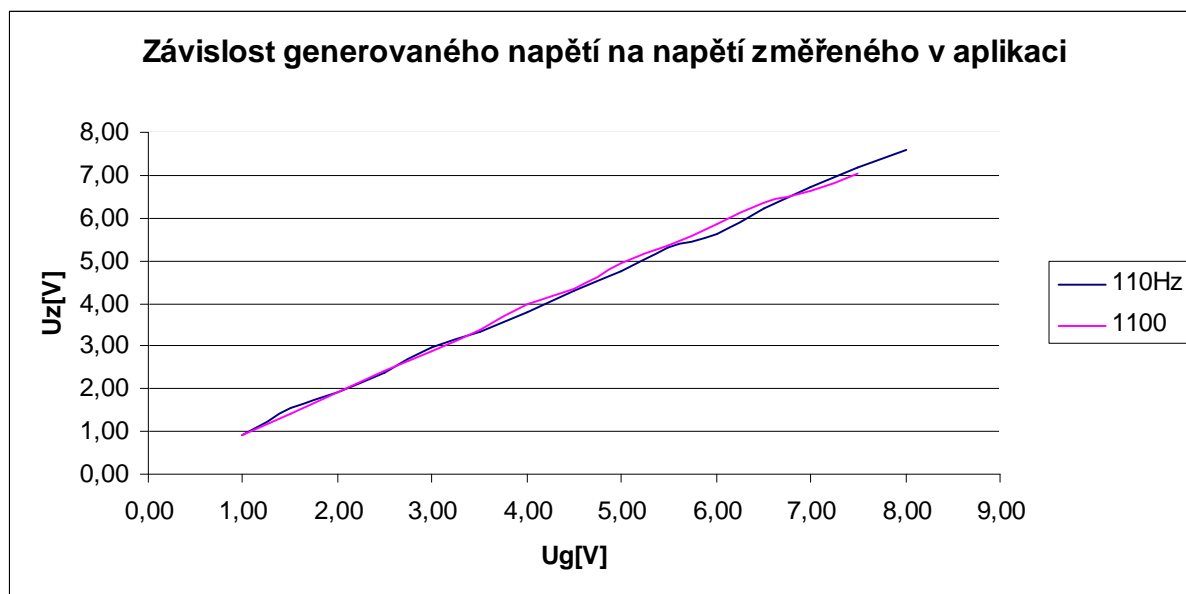
Dosažením do vzorců č. 11 a 12 se dopočítaly odchylky pro hodnoty druhého řádku s frekvencí 110Hz. tabulky č. 4.

$$U_o = 1,56 - 1,50$$

$$U_o = 0,06V$$

$$U_{o\%} = \frac{0,06}{1,50} \cdot 100$$

$$U_{o\%} = 4\%$$



Obr. 26 Graf závislosti generovaného napětí na napětí změřeného v aplikaci

## 7. Shrnutí výsledků a závěr

Hlavním účelem této práce byl návrh a realizace výukové aplikace pro zpracování signálů v programovém prostředí Matlab. Tato aplikace by měla sloužit studentům k pochopení probíraného učiva. Při vytváření aplikace jsem použil poznatků týkající se analýzy signálů. V aplikaci byla použita zvuková a měřicí karta. Díky tomu není aplikace omezena jen na simulování signálů, ale lze zde také nasimulovaný signál generovat či zobrazit vstupní signál.

Nejvyšší odchylky napětí byly vypočítány při nastavení menších hodnot napětí. Bylo to dáno velkým rušením signálu.

Při generování signálu pomocí zvukové karty se muselo nejprve vypočítat zesílení zvukové karty. Toto zesílení je typické pro určitou zvukovou kartu. Z toho vyplývá, že ne ve všech počítačích bude mít generovaný signál stejnou maximální hodnotu jako je maximální hodnota zadaná v aplikaci.

Další rozšíření aplikace bych viděl v analýze signálů s diskrétním čase nebo podporu více vstupně – výstupních zařízení.



## Literatura:

- [1] *Matlab* [online]. c1991-2009 [cit. 2009-01-23]. Dostupný z WWW:  
<<http://www.humusoft.cz/produkty/matlab/matlab/index.php?lang=cz&p1=1&p2=1&p3=1>>.
- [2] *Matlab* [online]. 2002- [cit. 2009-01-23]. Dostupný z WWW:  
<<http://cs.wikipedia.org/wiki/MATLAB>>.
- [3] *Simulink* [online]. 2002 [cit. 2009-01-25]. Dostupný z WWW:  
<<http://en.wikipedia.org/wiki/Simulink>>.
- [4] ZAPLATÍLEK, Karel, DOŇAR, Bohuslav. *Matlab : Tvorba uživatelských aplikací*. Praha : Ben, 2004. 216 s. ISBN 8073001330.
- [5] NEVŘIVA, Pavel. *Analýza signálů a soustav*. 1. vyd. Praha : Ben, 2000. 671 s. ISBN 80-7300-004-0.
- [6] PÍRKO, Zdeněk. *Fourierova transformace 1 : Užití v teoretické elektrotechnice*. 1. vyd. [s.l.] : Státní nakladatelství technické literatury, 1963. 84 s.
- [7] *MF 624* [online]. c1991-2009 [cit. 2009-04-02]. Dostupný z WWW:  
<<http://www.humusoft.cz/produkty/datacq/mf624/index.php?lang=cz&p1=1&p2=6&p3=1>>.
- [8] *Zásuvné desky do PC pro automatizaci a řízení* [online]. 2004 [cit. 2009-02-04]. Dostupný z WWW: <<http://www.automatizace.cz/article.php?a=1456>>.
- [9] *Real Time Toolbox* [online]. 1991-2009 [cit. 2009-02-25]. Dostupný z WWW:  
<<http://www.humusoft.cz/produkty/rtt/index.php?lang=cz&p1=1&p2=5>>.
- [10] *Alc 260* [online]. 2009 [cit. 2009-04-13]. Dostupný z WWW:  
<<http://www.realtek.com.tw/products/productsView.aspx?Langid=1&PFid=27&Level=5&Conn=4&ProdID=39>>

**Seznam příloh:**

Příloha č. 1: Zdrojový kód aplikace vytvořený v Matlabu

Příloha č. 2: Originální manuál od firmy Humusoft k měřicí kartě MF 624 ( mf624um.pdf)

Příloha č. 3: Originální manuál k osciloskopu Agilent řady 3000 (Agilent\_series\_3000.pdf)

Příloha č. 4: Originální manuál ke generátoru Agilent 33220A (Agilent\_33220A.pdf)

Příloha č. 5: Demonstrační video (Aplikace.avi)

Přílohy jsou přiloženy v elektronické podobě na CD.